Overview

• Motivation
• MapReduce/Hadoop in a nutshell
• Experimental cluster hardware example
• Application areas at the Austrian National Library
  • Web Archiving
  • Austrian Books Online
• SCAPE at the Austrian National Library
  • Hardware set-up
  • Open source software architecture
• Application Scenarios

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Cultural Heritage and Big Data?

• Google Books Project
  • 2012: 20 million books scanned (approx. 7,000,000,000 pages)
  • [www.books.google.com](http://www.books.google.com)

• Europeana
  • 2012: 25 million digital objects
  • All metadata licensed CC-0
  • [www.europeana.eu/portal](http://www.europeana.eu/portal)
Cultural Heritage and Big Data?

- Hathi Trust
  - 3,721,702,950 scanned pages
  - 477 TBytes
  - [www.hathitrust.org](http://www.hathitrust.org)

- Internet Archive
  - 245 billion web pages archived
  - 10 PBytes
  - [www.archive.org](http://www.archive.org)
What can we expect?

- **Enumerate** 2012: only about 4% digitised so far
- Strong growth of born digital information

Source:

Source:
MapReduce/Hadoop in a nutshell

Map

Input data

Task 1
Input split 1
Record 1
Record 2
Record 3

Task 2
Input split 2
Record 4
Record 5
Record 6

Task 3
Input split 3
Record 7
Record 8
Record 9

Reduce

Merge
Shuffle
Sort

Aggregated Result

Output data

Aggregated Result

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Experimental Cluster

CPU: 1 x 2.53GHz Quadcore CPU (8 HyperThreading cores)
RAM: 16GB
DISK: 2 x 1TB DISks configured as RAID0 (performance) – 2 TB effective

• Of 16 HT cores: 5 for Map; 2 for Reduce; 1 for operating system.

→ 25 processing cores for **Map** tasks and
→ 10 cores for **Reduce** tasks
Platform Architecture

Access via REST API
Workflow engine for complex jobs
Hive as the frontend for analytic queries
MapReduce/Pig for Extraction, Transform, and Load (ETL)
„Small“ objects in HDFS or HBase
„Large “ Digital objects stored on NetApp Filer

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Application scenarios

- **Web Archiving**
  - Scenario 1: Web Archive Mime Type Identification

- **Austrian Books Online**
  - Scenario 2: Image File Format Migration
  - Scenario 3: Comparison of Book Derivatives
  - Scenario 4: MapReduce in Digitised Book Quality Assurance
Key Data Web Archiving

Domain harvesting
Entire **top-level-domain .at** every 2 years

Selective harvesting
Important websites that change regularly

Event harvesting
Special occasions and events (e.g. elections)

Physical storage 19 TB
Raw data 32 TB
Number of objects 1,241,650,566
Scenario 1: Web Archive MIME Type Identification

(W)ARC Container

(W)ARC InputFormat

(W)ARC RecordReader

based on HERITRIX Web crawler
read/write (W)ARC

MapReduce

Apache Tika
detect MIME

Map

Reduce

- image/jpg: 1
- image/gif: 1
- text/html: 2
- audio/midi: 1

Tool integration pattern | Throughput (GB/min)
--- | ---
TIKA detector API call in Map phase | 6.17 GB/min
FILE called as command line tool from map/reduce | 1.70 GB/min
TIKA JAR command line tool called from map/reduce | 0.01 GB/min

Amount of data | Number of ARC files | Throughput (GB/min)
--- | --- | ---
1 GB | 10 x 100 MB | 1.57 GB/min
2 GB | 20 x 100 MB | 2.5 GB/min
10 GB | 100 x 100 MB | 3.06 GB/min
20 GB | 200 x 100 MB | 3.40 GB/min
100 GB | 1000 x 100 MB | 3.71 GB/min
Scenario 1: Web Archive Mime Type Identification

DROID 6.01

TIKA 1.0
Public private partnership with Google
Only public domain
Objective to scan ~ 600,000 Volumes
~ 200 Mio. pages
~ 70 project team members
20+ in core team
~ 130K physical volumes scanned so far
~ 40 Mio pages
This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Scenario 2: Image file format migration

- Task: Image file format migration
  - TIFF to JPEG2000 migration
    - Objective: Reduce storage costs by reducing the size of the images
  - JPEG2000 to TIFF migration
    - Objective: Mitigation of the JPEG2000 file format obsolescence risk

- Challenges:
  - Integrating validation, migration, and quality assurance
  - Computing intensive quality assurance
Scenario 2: Comparison of book derivatives

- Task: Compare different versions of the same book
  - Images have been manipulated (cropped, rotated) and stored in different locations
  - Images come from different scanning sources or were subject to different modification procedures

- Challenges:
  - Computing intensive (Average runtime per book on a single quad-core server ~ 4,5 hours)
  - 130,000 books, ~320 pages each

- SCAPE tool: Matchbox
Scenario 3: MapReduce in Quality Assurance

- ETL Processing of 60,000 books, ~ 24 Million pages
- Using Taverna’s „Tool service“ (remote ssh execution)
- Orchestration of different types of hadoop jobs
  - Hadoop-Streaming-API
  - Hadoop Map/Reduce
  - Hive
- Workflow available on myExperiment: http://www.myexperiment.org/workflows/3105
Scenario 3: MapReduce in Quality Assurance

Create input text files containing file paths (JP2 & HTML)

Read image metadata using Exiftool (Hadoop Streaming API)

Create sequence file containing all HTML files

Calculate average block width using MapReduce

Load data in Hive tables

Execute SQL test query
Reading image metadata

Jp2PathCreator

reading files from NAS

/HAS/Z119585409/00000001.jp2
/HAS/Z119585409/00000002.jp2
/HAS/Z119585409/00000003.jp2
...
/HAS/Z117655409/00000001.jp2
/HAS/Z117655409/00000002.jp2
/HAS/Z117655409/00000003.jp2
...
/HAS/Z119585987/00000001.jp2
/HAS/Z119585987/00000002.jp2
/HAS/Z119585987/00000003.jp2
...
/HAS/Z119584539/00000001.jp2
/HAS/Z119584539/00000002.jp2
/HAS/Z119584539/00000003.jp2
...
/HAS/Z119599879/00000001.jp2
/HAS/Z119599879/00000002.jp2
/HAS/Z119599879/00000003.jp2
...

1,4 GB

HadoopStreamingExiftoolRead

Z119585409/00000001  2345
Z119585409/00000002  2340
Z119585409/00000003  2543
...
Z117655409/00000001  2300
Z117655409/00000002  2300
Z117655409/00000003  2345
...
Z119585987/00000001  2300
Z119585987/00000002  2340
Z119585987/00000003  2432
...
Z119584539/00000001  5205
Z119584539/00000002  2310
Z119584539/00000003  2134
...
Z119599879/00000001  2312
Z119589879/00000002  2300
Z119589879/00000003  2300
...

1,2 GB

60.000 books (24 Million pages):  ~ 5 h  + ~ 38 h =  ~ 43 h
SequenceFile creation

HtmlPathCreator

reading files from NAS

/NAS/Z119585409/00000707.html
/NAS/Z119585409/00000708.html
/NAS/Z119585409/00000709.html
...
/NAS/Z138682341/00000707.html
/NAS/Z138682341/00000708.html
/NAS/Z138682341/00000709.html
...
/NAS/Z178791257/00000707.html
/NAS/Z178791257/00000708.html
/NAS/Z178791257/00000709.html
...
/NAS/Z967985409/00000707.html
/NAS/Z967985409/00000708.html
/NAS/Z967985409/00000709.html
...
/NAS/Z196545409/00000707.html
/NAS/Z196545409/00000708.html
/NAS/Z196545409/00000709.html
...

NAS

find

1,4 GB

60,000 books (24 Million pages): ~ 5 h

SequenceFileCreator

997 GB (uncompressed)

Z119585409/00000707
Z119585409/00000708
Z119585409/00000709
...
Z119585409/00000710
Z119585409/00000711
Z119585409/00000712

~ 29 h

60.000 books (24 Million pages): ~ 5 h + ~ 24 h = ~ 29 h
Calculate average block width using MapReduce

HadoopAvBlockWidthMapReduce

Map

Reduce

SequenceFile

60,000 books (24 Million pages):  ~ 6 h
Analytic Queries

HiveLoadExifData & HiveLoadHocrData

**CREATE TABLE htmlwidth**

<table>
<thead>
<tr>
<th>hid</th>
<th>hwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z119585409/00000001</td>
<td>1870</td>
</tr>
<tr>
<td>Z119585409/00000002</td>
<td>2100</td>
</tr>
<tr>
<td>Z119585409/00000003</td>
<td>2015</td>
</tr>
<tr>
<td>Z119585409/00000004</td>
<td>1350</td>
</tr>
<tr>
<td>Z119585409/00000005</td>
<td>1700</td>
</tr>
</tbody>
</table>

**CREATE TABLE jp2width**

<table>
<thead>
<tr>
<th>jid</th>
<th>jwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z119585409/00000001</td>
<td>2250</td>
</tr>
<tr>
<td>Z119585409/00000002</td>
<td>2150</td>
</tr>
<tr>
<td>Z119585409/00000003</td>
<td>2125</td>
</tr>
<tr>
<td>Z119585409/00000004</td>
<td>2125</td>
</tr>
<tr>
<td>Z119585409/00000005</td>
<td>2250</td>
</tr>
</tbody>
</table>
## Analytic Queries

### HiveSelect

#### jp2width

<table>
<thead>
<tr>
<th>jid</th>
<th>jwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z119585409/00000001</td>
<td>2250</td>
</tr>
<tr>
<td>Z119585409/00000002</td>
<td>2150</td>
</tr>
<tr>
<td>Z119585409/00000003</td>
<td>2125</td>
</tr>
<tr>
<td>Z119585409/00000004</td>
<td>2125</td>
</tr>
<tr>
<td>Z119585409/00000005</td>
<td>2250</td>
</tr>
</tbody>
</table>

#### htmlwidth

<table>
<thead>
<tr>
<th>hid</th>
<th>hwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z119585409/00000001</td>
<td>1870</td>
</tr>
<tr>
<td>Z119585409/00000002</td>
<td>2100</td>
</tr>
<tr>
<td>Z119585409/00000003</td>
<td>2015</td>
</tr>
<tr>
<td>Z119585409/00000004</td>
<td>1350</td>
</tr>
<tr>
<td>Z119585409/00000005</td>
<td>1700</td>
</tr>
</tbody>
</table>

```sql
select jid, jwidth, hwidth from jp2width inner join htmlwidth on jid = hid
```
Further information

• Project website: www.scape-project.eu
• Github repository: www.github.com/openplanets
• Project Wiki: www.wiki.opf-labs.org/display/SP/Home

SCAPE tools mentioned

• SCAPE Platform
  • http://www.scape-project.eu/publication/an-architectural-overview-of-the-scape-preservation-platform
• Jpylyzer – Jpeg2000 validation
  • http://www.openplanetsfoundation.org/software/jpylyzer
• Matchbox – Image comparison
  • https://github.com/openplanets/scape/tree/master/pc-qa-matchbox

Thank you! Questions?